BEYONDRETARGET: LEARNING EXECUTABLE HUMANOID MOTIONS DIRECTLY FROM MONOCULAR VIDEO

Anonymous authors
Paper under double-blind review

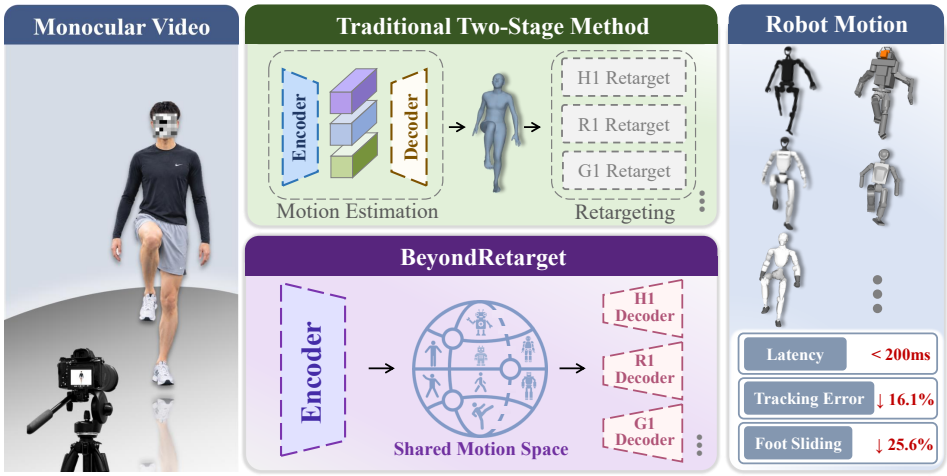


Figure 1: BeyondRetarget encodes the video into a shared motion space and directly decodes robot-specific trajectories for the requested humanoids.

ABSTRACT

Learning executable motions from human videos offers a scalable solution for humanoid robots to acquire demonstration motions. However, existing pipelines typically first construct an explicit human motion representation and then convert it into robot motions via motion retargeting. Although such methods can effectively leverage large volumes of existing human data for training, the substantial differences between humans and humanoid robots in locomotion mechanisms and joint degree-of-freedom configurations make motions generated by this human-representation-centric approach difficult to execute on robots. Furthermore, errors introduced during human motion estimation inevitably propagate to the retargeting stage and cannot be eliminated via joint optimization. We propose BeyondRetarget, an end-to-end framework that directly maps monocular RGB videos to robot motions. Discarding the explicit human representation, this framework learns robot-oriented implicit representations directly from visual observations, enabling the model to capture cross-morphology motion structures. To generate motions more suitable for robot execution, we further design a contact-aware motion optimization mechanism to improve temporal consistency and physical plausibility. Experiments show that BeyondRetarget significantly improves the accuracy and robustness of generated robot motions, while achieving higher execution success rates and lower latency in both simulation environments and real humanoid robots.

1 INTRODUCTION

Large-scale, executable motion data serve as a fundamental foundation for humanoid robots to perform skill learning and whole-body control. However, such data is typically acquired using specialized systems such as optical motion capture (Vicon Motion Systems Ltd., 2026) and wearable inertial sensors (Schepers et al., 2010), which require dedicated hardware and controlled capture en-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

vironments, leading to high costs and difficulties in large-scale deployment (Schepers et al., 2010). In contrast, monocular RGB videos are abundant and low-cost to collect. Therefore, monocular videos represent a highly promising data source for large-scale acquisition of robot motion data.

Benefiting from recent advances in human motion capture techniques, monocular RGB videos can be converted into unified explicit human motion representations (Shin et al., 2024; Shen et al., 2024), such as Skinned Multi-Person Linear Model (SMPL) (Loper et al., 2015). Built upon this mature human motion representation, existing video-based robot motion generation methods typically adapt a two-stage pipeline: reconstructing human motion from RGB videos first, then retargeting human motion onto the target robot (Fu et al., 2024; Allshire et al., 2025; Roux et al., 2026). This pipeline fully leverages the high similarity in overall structure and motion patterns between humans and humanoid robots, enabling extraction and transfer of meaningful motion patterns from human movements, and thus achieves promising results in most scenarios.

However, the similarity between humans and humanoid robots mainly lies in global morphology and kinematic structure, while notable discrepancies remain in body proportions, joint degrees of freedom (DoF), motion dynamics and other aspects. Furthermore, the two-stage decomposition propagates human motion reconstruction errors from the first stage directly into the subsequent retargeting process, which compound with retargeting errors and form error accumulation that cannot be eliminated via joint optimization. Consequently, although the two-stage method provides a natural and effective solution for video-to-robot motion generation, its human-centric intermediate representation limits the capability to directly model motion within the robot space.

To address these issues, we propose BeyondRetarget, an end-to-end framework that directly generates executable humanoid robot motion from monocular RGB videos. The core idea is to eliminate the explicit SMPL intermediate representation and instead learn an implicit, robot-oriented motion representation directly from visual observations. This representation can be optimized directly for cross-morphology target robot motion generation. Furthermore, it can be mapped to motions of any target humanoid robot via robot-specific decoders. We establish geometric correspondences between humans and robots via a set of semantic keypoints and semantic link directions, and constructs a unified supervision mechanism to directly supervise robot motion using SMPL-based data. To improve motion executability, a contact-aware optimization module identifies foot contact states, determines support states, and performs kinematic corrections to enhance temporal consistency and physical plausibility. Figure 1 illustrates the core modifications brought by this work.

Experiments demonstrate that BeyondRetarget can generate accurate, robust, and executable humanoid robot motions from monocular videos. Compared with conventional two-stage pipelines, BeyondRetarget achieves lower pose errors and substantially reduces severe motion collapse, while more faithfully reproducing human motions and maintaining stable execution in both simulation and real-robot settings. Moreover, the shared motion features and unified supervision scheme enable efficient adaptation to diverse humanoid embodiments. These results demonstrate the potential of BeyondRetarget to transform the abundant and easily accessible monocular videos into high-quality executable robot motion data, providing a scalable and low-cost approach for large-scale robot motion data acquisition.

Our main contributions are:

- We propose an end-to-end framework that **directly generates executable humanoid robot motion from monocular human videos**, providing a convenient solution for acquiring a large-scale data for humanoid robots.
- We propose an **implicit, shared robot-oriented motion representation**, together with a unified human-robot supervision mechanism, enabling this shared representation to be decoded onto different humanoid robot embodiments.
- We introduce explicit contact and support representation to refine the estimated motion for improved temporal and **physical consistency**.
- We systematically **evaluate our method in both simulation and real-world robot platforms**, showing superior motion accuracy, robustness, execution success rate and real-time performance.

2 Related Work

Existing methods for generating robot motion data typically follow a two-stage pipeline (Fu et al., 2024; Allshire et al., 2025; Roux et al., 2026), first recovering explicit human motion and then retargeting it to robots. This paper reviews related work from two aspects: human motion data acquisition and representation, and human-to-robot motion retargeting.

Human motion acquisition and representation. Traditional systems capture human motion by tracking optical markers (Vicon Motion Systems Ltd., 2026) or fusing wearable inertial and magnetic measurements (Schepers et al., 2010). The captured motion is represented as marker trajectories (Loper et al., 2014) or skeletal joint positions and rotations (Ionescu et al., 2014; Huang et al., 2018). These systems provide detailed measurements but require dedicated hardware and venues, limiting collection at scale. Parametric body models subsequently link skeletal motion to body shape and surface deformation (Anguelov et al., 2005). MoSh fits such models to sparse markers (Loper et al., 2014), while SMPL provides a shared mesh topology and skeletal structure (Loper et al., 2015). SMPL's consistent pose and shape parameters offer a common interface for learning and retargeting across heterogeneous data sources. MANO and SMPL-X extend this representation to articulated hands and expressive bodies (Romero et al., 2017; Pavlakos et al., 2019). AMASS unifies 15 optical motion-capture datasets in a common SMPL-based representation (Mahmood et al., 2019), while BEDLAM uses synthetic bodies and motion to scale image-based supervision (Black et al., 2023). MotionPRO and RICH further connect motion to pressure and scene contact (Ren et al., 2025; Lu et al., 2026; Huang et al., 2022).

With these parametric body models providing a consistent representation of human motion, monocular RGB videos have emerged as an attractive source for scalable motion acquisition due to their abundance and ease of collection without specialized hardware. Single-image estimation evolved from model fitting to learned regression (Bogo et al., 2016; Kanazawa et al., 2018; Goel et al., 2023). Recovering motion, however, requires consistency across frames. Video methods therefore introduced temporal encoding and learned motion priors (Kanazawa et al., 2019; Kocabas et al., 2020; Luo et al., 2020), followed by stronger temporal context and attention mechanisms (Choi et al., 2021; Wan et al., 2021). WHAM further refines trajectories using contact information (Shin et al., 2024), while GVHMR introduces gravity-view coordinates to reduce global recovery ambiguity (Shen et al., 2024). Together, these developments expand the data sources for motion acquisition, enabling monocular videos to be converted into structured human motion data, which can be transformed into corresponding robot action data through a retargeting pipeline.

Human-to-robot motion retargeting. Retargeting transfers human motion to robots with different kinematics while preserving motion characteristics under robot constraints. Early approaches establish **hand-designed** correspondences between human and robot joints, followed by motion scaling, inverse kinematics, and trajectory modification (Pollard et al., 2002; Nakaoka et al., 2003; Ude et al., 2004). These methods establish the basic human-to-robot motion generation pipeline but require substantial robot-specific design. More recently, **optimization-based** methods formulate retargeting as constrained motion matching, jointly considering motion fidelity, morphology, and feasibility (Liarokapis et al., 2013; Ayusawa & Yoshida, 2017; Darvish et al., 2019). GMR (Araujo et al., 2025) is a representative recent approach that further improves motion fidelity and addresses physical artifacts. However, such non-convex optimization can be sensitive to initialization and objective design, leading to local collapse or discontinuous joint trajectories (Zhao et al., 2026). To avoid these numerical instabilities, **learning-based** methods directly learn mappings from human to robot motion (Villegas et al., 2018; Aberman et al., 2020). NMR (Zhao et al., 2026), for example, learns a robot motion mapping from high-quality paired supervision, improving inference stability and smoothness. However, the learned mapping is coupled to the target robot and training distribution, making adaptation to new robot morphologies costly. Another line of work (Luo et al., 2026; Li et al., 2025) maps human motion representations to latent commands for direct use in downstream control. While such latent interfaces reduce reliance on explicit retargeting, they provide less interpretable robot motion and limited access to complete trajectories for inspection and editing.

Overall, explicit robot motion remain a high-quality and indispensable source of data for robot motion learning and imitation. However, existing methods for acquiring such data typically decouple human motion capture from robot motion retargeting, resulting in inherent limitations such as information bottlenecks and error accumulation that limit the quality of generated motion.

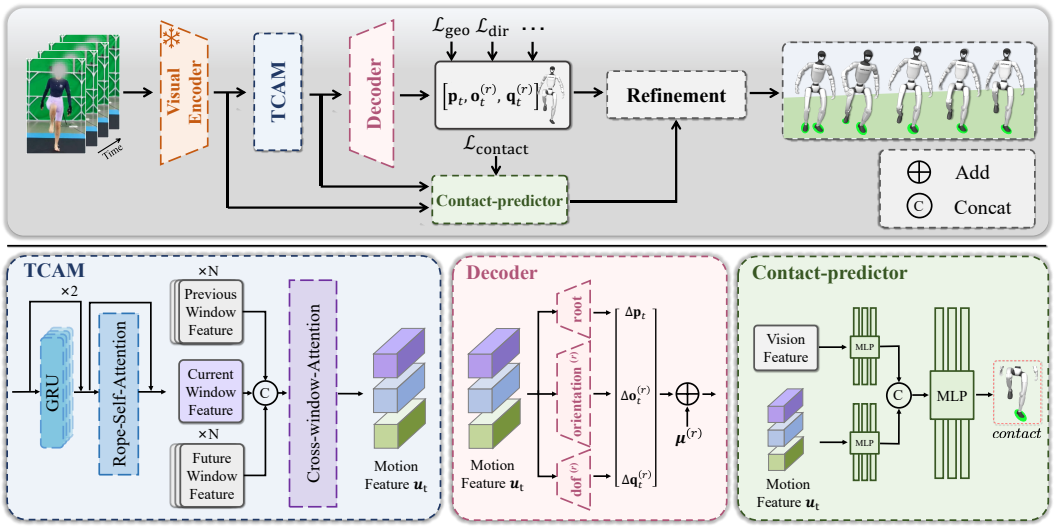


Figure 2: Model architecture. Input RGB video is encoded by the visual and temporal encoders, and the resulting shared motion representation is directly decoded into the target robot motion. Contact-aware refinement further improves support and temporal consistency.

3 METHOD

3.1 PROBLEM FORMULATION AND FRAMEWORK OVERVIEW

Given a monocular RGB video sequence $\mathcal{I} = \{\mathbf{i}_t\}_{t=1}^{n}$ of length n and a target robot configuration r, our goal is to directly estimate the complete motion of the target robot:

$$\hat{\mathcal{M}}^{(r)} = \{\hat{\mathbf{p}}_t, \hat{\mathbf{o}}_t^{(r)}, \hat{\mathbf{q}}_t^{(r)}\}_{t=1}^{n}, \tag{1}$$

where $\hat{\mathbf{p}}_t \in \mathbb{R}^3$ denotes the root translation, while $\hat{\mathbf{o}}_t^{(r)} \in \mathbb{R}^6$ and $\hat{\mathbf{q}}_t^{(r)} \in \mathbb{R}^{d_r}$ denote the root orientation and the d_r-dimensional DoF vector of target robot r, respectively.

As illustrated in Figure 2, we use a frozen visual encoder (Goel et al., 2023) to encode the video into visual features, which are then temporally modeled by the Temporal Context Aggregation Module (TCAM) (Lu et al., 2026) to produce a shared motion representation $\mathbf{u}_t$. This representation bridges the motion spaces of human and humanoid and can serve as a shared input for motion decoders of different robots.

For a target robot r, the robot motion decoder Ψ_r directly maps the shared motion representation to robot motion. We parameterize the motion as a residual around a robot-specific nominal state:

$$[\hat{\mathbf{p}}_t, \hat{\mathbf{o}}_t^{(r)}, \hat{\mathbf{q}}_t^{(r)}] = \boldsymbol{\mu}^{(r)} + [\Delta\mathbf{p}_t, \Delta\mathbf{o}_t^{(r)}, \Delta\mathbf{q}_t^{(r)}], \tag{2}$$

where $[\Delta\mathbf{p}_t, \Delta\mathbf{o}_t^{(r)}, \Delta\mathbf{q}_t^{(r)}] = \Psi_r(\mathbf{u}_t)$ denotes the residuals for root translation, root orientation, and robot-specific joint motion, respectively, and $\boldsymbol{\mu}^{(r)} \in \mathbb{R}^{3+6+d_r}$ is a learnable nominal state of robot r.

The decoder consists of an MLP trunk, a root translation branch, a root orientation branch, and a robot-specific Dof branch. For each additional robot, the pretrained visual temporal representation, MLP trunk, and root translation branch are reused, while its Dof and root orientation branch are adapted to the robot's kinematic structure and degrees of freedom.

3.2 UNIFIED MOTION SUPERVISION AND TRAINING

Human pose parameters and robot's DoF differ in dimension and kinematic meaning, making supervision infeasible during training. In order to directly use human body annotations for training, we introduce a non-uniform scaling method that establishes a shared set of keypoints between human and robot, and adapts SMPL to the morphology of the target humanoid robot, thereby transferring motion supervision to a unified geometric space.

Constructing the unified motion supervision. Given that all humanoid robots and the human body have similar topological structures, we define a set $\mathcal{J}$ of 18 shared keypoints covering the pelvis, torso, shoulders, elbows, wrists, hands, hips, knees, ankles, and feet. On the human side, these keypoints are extracted from the SMPL joints, denoted as $\mathcal{A} = \{\mathbf{a}_{t,j} \in \mathbb{R}^3\}$ where $t \in \{1, \ldots, n\}$ is the frame index and $j \in \mathcal{J}$ is the keypoint index. On the robot side, keypoints $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_{t,j} \in \mathbb{R}^3\}$ are a series of corresponding joints.

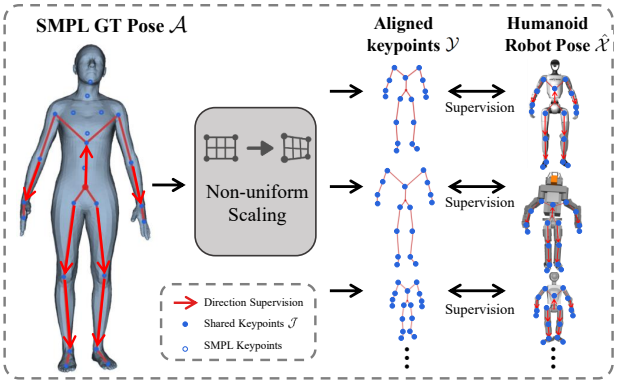


Figure 3: Unified human and robot supervision. Human keypoints are adapted to robot morphology. Keypoint positions and directed segments supervise humanoid robot pose.

To align the morphology of the human and the robot, we scale each link of the skeleton independently. Let $\pi(j)$ denote the parent of keypoint j, and let $\ell_j^{(\mathrm{h})}$ and $\ell_j^{(\mathrm{r})}$ denote the lengths of link $(\pi(j), j)$ for the human reference and the target robot, respectively. Taking the pelvis keypoint 0 as the root, we construct the robot-proportioned target as

$$\mathbf{y}_{t,0} = \mathbf{a}_{t,0}, \qquad \mathbf{y}_{t,j} = \mathbf{y}_{t,\pi(j)} + \frac{\ell_j^{(\mathrm{r})}}{\ell_j^{(\mathrm{h})}} \left(\mathbf{a}_{t,j} - \mathbf{a}_{t,\pi(j)}\right). \tag{3}$$

The resulting $\mathbf{y}_{t,j}$ forms the robot-proportioned supervision target. Human annotations can therefore directly supervise robot motions without native robot trajectories.

For each frame, we align the estimated robot keypoint set $\hat{\mathcal{X}}$ to the scaled human keypoint set $\mathcal{Y} = \{\mathbf{y}_{t,j} \in \mathbb{R}^3\}$ via the Procrustes transformation ϕ_t. Let $\tilde{\mathbf{x}}_{t,j} = \phi_t(\hat{\mathbf{x}}_{t,j}, \mathbf{y}_{t,j})$ denote the robot keypoint after alignment. The geometry loss is

$$\mathcal{L}_{\mathrm{geo}} = \frac{1}{n|\mathcal{J}|} \sum_{t=1}^{n} \sum_{j \in \mathcal{J}} \rho(\tilde{\mathbf{x}}_{t,j} - \mathbf{y}_{t,j}), \tag{4}$$

where ρ denotes the coordinate-wise Smooth L_1 penalty and $|\cdot|$ denotes the cardinality of a set.

Keypoint positions do not fully constrain local orientation, so we define a set $\mathcal{D}$ of directed semantic links corresponding to the upper arms, forearms, thighs, shanks, and torso. For $d \in \mathcal{D}$, let $\hat{\mathbf{v}}_{t,d}$ denote the normalized estimated link vectors computed by $\hat{\mathbf{x}}_{t,j}$, and $\mathbf{v}_{t,d}$ denote the target link vectors computed by $\mathbf{y}_{t,j}$, respectively. The direction loss penalizes their cosine discrepancy:

$$\mathcal{L}_{\mathrm{dir}} = \frac{1}{n|\mathcal{D}|} \sum_{t=1}^{n} \sum_{d \in \mathcal{D}} \left(1 - \hat{\mathbf{v}}_{t,d}^{\top} \mathbf{v}_{t,d}\right). \tag{5}$$

This constraint can effectively improve the accuracy of the link direction and make the robot's motion more natural and realistic.

Training objective and multiple robots. The complete training loss combines geometry, direction, temporal consistency, and robot joint limits:

$$\mathcal{L}_{\mathrm{motion}} = \lambda_{\mathrm{geo}}\mathcal{L}_{\mathrm{geo}} + \lambda_{\mathrm{dir}}\mathcal{L}_{\mathrm{dir}} + \lambda_{\mathrm{trans}}\mathcal{L}_{\mathrm{trans}} + \lambda_{\mathrm{ori}}\mathcal{L}_{\mathrm{ori}} + \lambda_{\mathrm{temp}}\mathcal{L}_{\mathrm{temp}} + \lambda_{\mathrm{lim}}\mathcal{L}_{\mathrm{lim}}. \tag{6}$$

Each $\lambda_{\{\cdot\}}$ denotes the weight for its corresponding loss term. $\mathcal{L}_{\mathrm{geo}}$ and $\mathcal{L}_{\mathrm{dir}}$ supervise the relative robot pose within the unified motion space. $\mathcal{L}_{\mathrm{trans}}$ and $\mathcal{L}_{\mathrm{ori}}$ constrain the root translation and rotation motion, respectively. $\mathcal{L}_{\mathrm{temp}}$ enforces temporal consistency across consecutive frames, and $\mathcal{L}_{\mathrm{lim}}$ penalizes violations of robot joint limits.

We first train the encoder and complete robot motion decoder on Unitree G1 robot. For a new robot, we configure its kinematic tree, semantic keypoints, and morphology scales, reuse the encoder, TCAM and root translation branch in decoder, and only train its robot-specific Dof and root orientation branch.

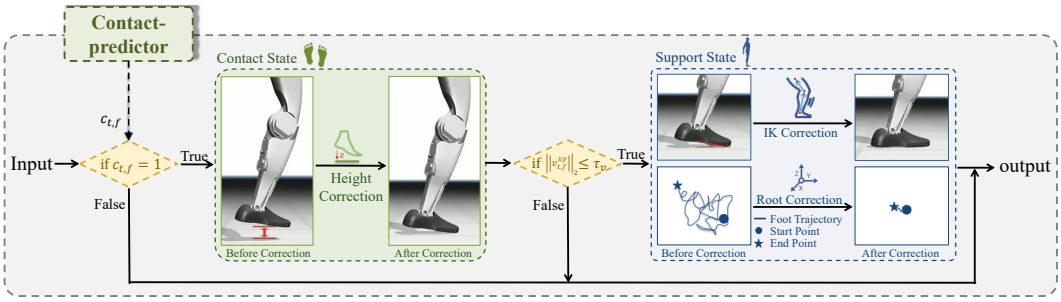


Figure 4: Contact-aware kinematic refinement. The robot's lower limb are divided into contact or support states based on explicit contact representation, and optimized via corresponding post-processing to improve the physical plausibility.

3.3 Contact-Aware Refinement

The estimator is capable of generating complete robot motion. However, accurate poses inferred from visual inputs do not necessarily yield physically executable motions. This is because the executability of robot motion largely depends on temporal consistency and contact stability, whereas the unified motion supervision scheme provides insufficient constraints on these two properties. To address this issue, we first freeze the motion estimator, filter its generated motions, then introduce explicit contact representations to guide subsequent motion optimization.

Specifically, after motion training, we fix estimator's parameters and train a contact predictor under pressure data supervision (Ren et al., 2025). This predictor fuses the motion feature with the visual feature to predict the contact probability of each foot. As illustrated in Figure 4, we use a two-level state decision. First, we convert the predicted contact probability $\hat{c}_{t,f} \in [0, 1]$ of each foot f into a binary contact state $c_{t,f} \in \{0, 1\}$, where 0 denotes no contact and 1 denotes contact. We then obtain the foot trajectory and its horizontal velocity $v_{t,f}^{xy}$ through forward kinematics. A contacted foot is regarded as being in the support state only when its horizontal velocity is below the threshold τ_v:

$$s_{t,f} = c_{t,f} \wedge \left(\left\| v_{t,f}^{xy} \right\|_2 \leq \tau_v \right). \tag{7}$$

$||\cdot||_2$ denotes the vector norm. This distinction separates feet that are in contact with the ground but still moving from those providing stable support. The contact state triggers root height correction: we eliminate height offsets to prevent foot penetration or floating. The support state additionally enables root horizontal correction and lower-body inverse kinematics (IK) solving to ensure no sliding of stably supporting feet. This hierarchical structure maintains stable support feet while naturally allowing sliding for moving contact feet. To avoid abrupt changes caused by support state switching, we apply a temporal smoothing filter $\mathrm{Filt}(\cdot)$ to the residual sequence and add the filtered correction to the original prediction to obtain the final refined joint posture. Let $\hat{\mathbf{q}}_t$ denote the raw predicted full-body joint posture at frame t, and $\mathbf{q}_t^{\mathrm{IK}}$ denote the optimized joint posture solved via constrained inverse kinematics. The IK correction residual across all frames $t \in \{1, \ldots, n\}$ is formulated as:

$$\Delta \mathbf{q}_t^{\mathrm{IK}} = \mathbf{q}_t^{\mathrm{IK}} - \hat{\mathbf{q}}_t, \qquad \hat{\mathbf{q}}_t \leftarrow \hat{\mathbf{q}}_t + \mathrm{Filt}\left(\Delta \mathbf{q}_t^{\mathrm{IK}}\right). \tag{8}$$

The filtered correction ensures smooth and physically consistent motion transitions while maintaining accurate foot contact and support constraints.

4 Experiments

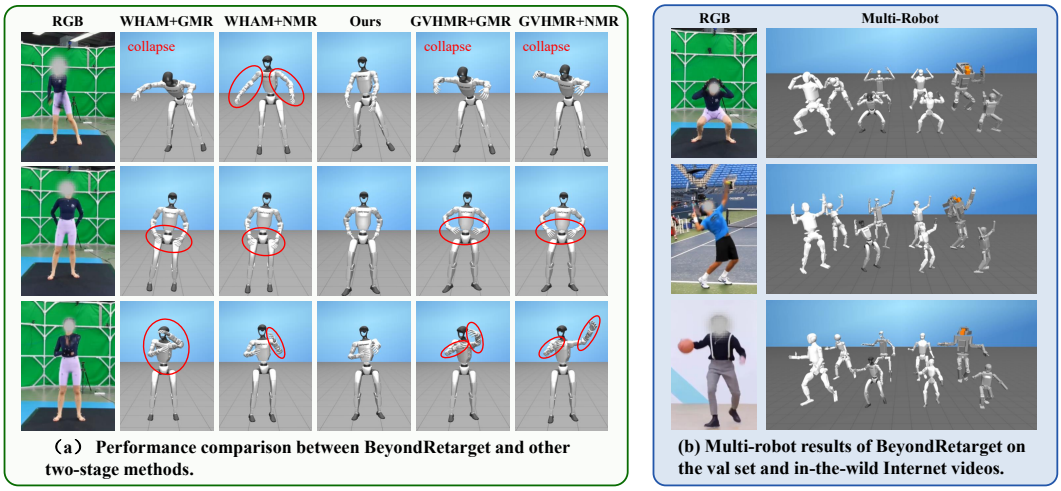
We organize the experiments around four questions: (1) Can our end-to-end method outperform the two-stage methods that first reconstruct human motion and then retarget it; (2) Can shared motion representation adapt to multiple humanoid robots; (3) How does each component affect model performance; (4) How does the motion output by our method perform on a real robot? Can it support real-time motion control?

4.1 Experimental Setup

Setup. We train on MotionPRO (Ren et al., 2025), which provides synchronized RGB images, optical motion capture SMPL ground truth (GT), and foot contact labels, covering approximately 12,000,000 frames and over 400 action categories. SMPL GT supervises robot motion, while foot

Table 1: Quantitative comparison on G1. The oracle rows at the top use SMPL GT to isolate errors from human motion reconstruction, and thus are excluded from the ranking.

Method	MAMPJPE ↓	RTE ↓	Jitter ↓	Accel ↓	Fail_{65} ↓	Fail_{100} ↓	FS ↓	Exec. SR ↑	Exec. MAMPJPE ↓
GT→NMR	33.26	837.62	1.71	1.16	3.99	1.39	1.30	346/362	40.32
GT→GMR	32.02	36.01	10.65	2.27	0.00	0.00	0.82	347/362	35.93
GVHMR→NMR	39.19	701.13	1.63	1.56	6.21	2.22	2.28	340/362	43.32
GVHMR→GMR	39.46	637.33	7.94	2.51	3.43	0.28	3.30	313/362	41.77
WHAM→NMR	43.15	849.36	4.87	3.07	9.36	4.17	6.64	342/362	46.56
WHAM→GMR	45.88	217.76	63.26	14.57	10.47	6.30	13.30	295/362	42.89
BeyondRetarget	26.36	80.74	1.41	1.51	0.00	0.00	1.71	347/362	35.04



(a) Performance comparison between BeyondRetarget and other two-stage methods.

(b) Multi-robot results of BeyondRetarget on the val set and in-the-wild Internet videos.

Figure 5: Qualitative results.

contact labels are used to train the contact predictor. We train the model on Unitree G1, and adapt each additional humanoid (Unitree R1, Fourier GR1-T1, Fourier GR2-V3, Unitree H1, Booster T1, Tienkung, and Atlas) by retraining its joint and root orientation branches in the robot-specific decoder. Adapting a new robot takes approximately one hour on a single RTX 4090. We compare against GVHMR (Shen et al., 2024)/WHAM (Shin et al., 2024) combined with NMR (Zhao et al., 2026)/GMR (Araujo et al., 2025). To isolate errors from human motion reconstruction, Oracle baseline uses SMPL GT for direct retargeting.

Metrics. We evaluate local pose, global trajectory, temporal continuity, physical plausibility and executability through the following metrics. Morphology-Aligned MPJPE (MAMPJPE, mm) measures the keypoint error after aligning the reference human body with the robot morphology, while Root Trajectory Error (RTE, mm) measures root trajectory error after rigid alignment. Torso Direction Error (TDE, °) and End-effector Direction Error (EDE, °) are used to measure the angular errors of the torso vector and the end-effector vector relative to the GT vector, respectively. Jitter (10 m/s^3) and Accel (mm/frame^2) measure higher order temporal jitter and acceleration error. Foot sliding (FS, mm/frame) measures the unintended horizontal displacement of the feet during support frames, while Violation rate (Viol., %) measures the proportion of contact frames exhibiting foot floating or ground penetration. Fail_{65} (%) and Fail_{100} (%) measure the proportions of windows whose average MAMPJPE exceeds 65 mm and 100 mm, respectively, over the total number of windows (each consists of 200 frames). Executed Success Rate (Exec. SR, %) measures the ratio of successfully executed clips to the total number of clips in physical simulation (Mujoco). We use a general motion tracker SONIC (Luo et al., 2026) to execute the output motion, and a motion clip (each consists of 600 frames) is considered failed if its execution is terminated before reaching the end of the reference motion due to excessive root height, root orientation, or end-effector height errors. Executed Morphology-Aligned MPJPE (Exec. MAMPJPE, mm) measures the accuracy of the executed motion.

4.2 Comparison with Two-Stage Methods

Table 1 compares pose, trajectory, temporal consistency, and physical plausibility. BeyondRetarget consistently outperforms all two-stage methods across MAMPJPE, RTE, Jitter, and Foot Slide,

Table 2: Multi-robot MAMPJPE evaluation. The "–" entries denote robots unsupported by GMR.

Method	H1	T1	Tienkung	G1	R1	GR1-T1	GR2-V3	Atlas
GVHMR→GMR	77.79	70.09	103.23	39.46	–	–	–	–
WHAM→GMR	99.66	77.26	115.86	45.88	–	–	–	–
BeyondRetarget	**51.32**	**51.25**	**67.99**	**26.36**	**27.81**	**43.78**	**49.06**	**72.05**

Table 3: Component ablation results.

Variant	MAMPJPE ↓	RTE ↓	TDE ↓	EDE ↓	Jitter ↓	Accel ↓	FS ↓	Viol. ↓
w/o non-uniform alignment	36.08	89.84	5.17	15.77	1.60	1.58	1.89	4.39
w/o $\mathcal{L}_{\mathrm{dir}}$	29.62	94.77	6.95	21.78	1.45	1.59	2.08	7.00
w/o $\mathcal{L}_{\mathrm{temp}}$	26.44	77.12	2.95	15.41	1.93	1.61	1.99	5.66
w/o refinement	25.73	91.33	2.95	14.31	2.17	1.74	4.51	22.18
Full model	26.36	80.74	2.95	15.29	1.42	1.51	1.71	6.17

demonstrating that end-to-end estimation improves motion accuracy, temporal consistency and physical plausibility. More importantly, BeyondRetarget eliminates severe pose collapse: no evaluated window exceeds either the 65 mm or 100 mm MAMPJPE threshold, while all two-stage methods exhibit such collapse. This indicates that removing the intermediate human representation not only improves average accuracy but also substantially enhances robustness. As illustrated in Figure 5(a), for large-amplitude motions such as arm swings, two-stage pipelines suffer from misplaced limbs and pose collapse, whereas the direct estimation approach preserves motion characteristics more faithfully. For fine-grained actions, such as the hands-behind-the-waist poses, all two-stage methods incorrectly place the hands in front of the body, whereas BeyondRetarget accurately reconstructs the hands-behind-the-back motion.

To isolate retargeting error from human estimation error, we evaluate an Oracle setting where SMPL GT is directly fed into the retargeting module. BeyondRetarget still achieves lower MAMPJPE than both Oracle retargeting baselines, demonstrating that the retargeting stage itself introduces non-negligible motion errors and providing further evidence for avoiding the intermediate human model.

Finally, in physical simulation, BeyondRetarget achieves a 347/362 (95.86%) success rate with a Exec. MAMPJPE of 35.04 mm, outperforming all two-stage pipelines using estimated human motion. Together, these results show that direct estimation improves RGB-to-robot motion accuracy, avoids pose collapse introduced by retargeting, and produces more executable reference motions.

4.3 Multi-robot experiments

Table 2 evaluates the same adaptation procedure on eight humanoids with different morphologies. NMR is not included because it currently supports only the Unitree G1 and adapting it to a new robot morphology requires corresponding robot data. Due to differences in joint structures, different robots have varying MAMPJPE values, but all falling within a reasonable range of 26.36 mm to 72.05 mm, and our method outperforms GVHMR→GMR on all four robots supported by both methods. Figure 5(b) further shows that all eight humanoids can reproduce the same visual action, with performance generalizing to in-the-wild Internet videos, which indicates that the unified supervision can transfer the shared visual motion representation across different robot embodiments and possesses strong visual generalization capability.

4.4 Ablation studies

Table 3 presents the metric results obtained by separately ablating non-uniform morphology alignment, $\mathcal{L}_{\mathrm{dir}}$, $\mathcal{L}_{\mathrm{temp}}$, and contact-aware refinement. First, removing non-uniform morphology alignment increases MAMPJPE from 26.36 mm to 36.08 mm, the largest pose degradation among the ablations, demonstrating non-uniform morphology alignment can provide more reasonable supervision. Removing $\mathcal{L}_{\mathrm{dir}}$ increases MAMPJPE, TDE and EDE, demonstrating that explicitly constraining semantic link directions enables more accurate local orientations of body parts.

Second, removing $\mathcal{L}_{\mathrm{temp}}$ substantially degrades Jitter, Accel, and FS, demonstrating its effectiveness in improving temporal continuity. Finally, removing contact-aware refinement substantially

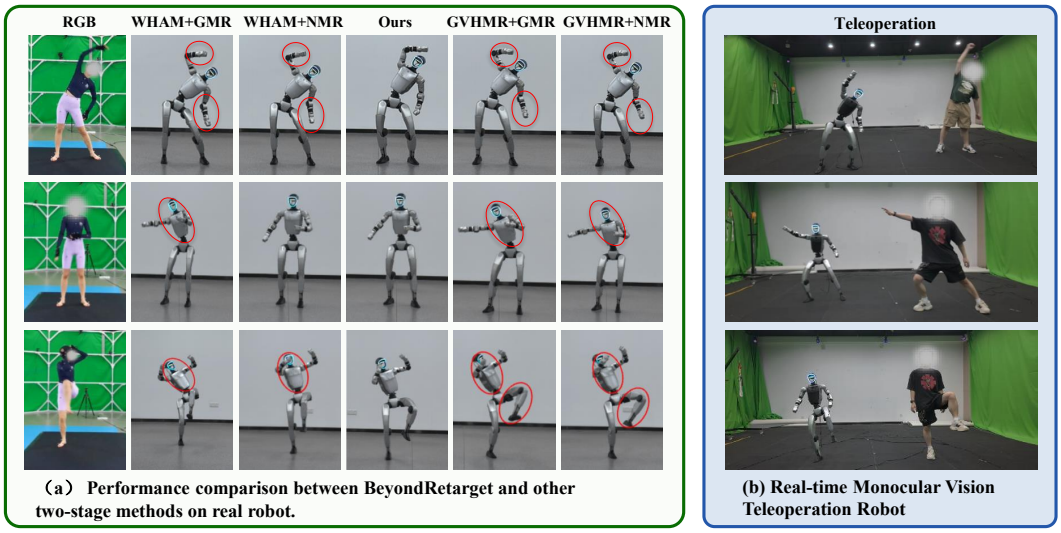


Figure 6: Real-robot evaluation and visual teleoperation.

degrades RTE, FS, and Viol. This shows that refinement corrects the raw trajectories and improves physical plausibility while preserving action accuracy.

4.5 Real-robot and teleoperation experiments

Figure 6(a) compares five methods executing the same input motions on a real robot. For the stretching motion, BeyondRetarget successfully extends the left arm above the head with the hand kept close to the thigh, whereas the four two-stage baselines exhibit clear arm motion discrepancies. For the single-arm circling and hands-on-head motion, BeyondRetarget maintains stable body motion and foot support throughout execution, while all four two-stage baselines show apparent instability. Together with the simulation results in Table 1, these results demonstrate that the improvements in motion accuracy and physical stability achieved in physical simulation transfer effectively to real-robot execution, further validating the superior executability of BeyondRetarget.

Table 4: Real-time performance comparison.

Method	latency (ms)	throughput (FPS)	VRAM (GiB)
GVHMR→GMR	1369.4	44.31	9.18
GVHMR→NMR	1305.7	44.36	9.73
BeyondRetarget	**192.8**	**50.02**	**7.03**

On the same GPU, we build streaming inference pipelines for different methods and configure each to achieve the lowest possible latency, then measure their median latency and peak VRAM usage. We separately test their maximum throughput without limiting the input frame rate. Table 4 shows that, benefiting from our end-to-end framework and streamlined model architecture, BeyondRetarget achieves lower output latency, reduced peak GPU memory usage, and higher maximum throughput compared to both two-stage baselines. This efficiency supports responsive real-time visual teleoperation, as demonstrated in Figure 6(b).

5 Conclusion and Limitations

Conclusion. We present BeyondRetarget, an end-to-end framework that generates executable humanoid robot motion from monocular human videos. Experiments demonstrate that our framework achieves higher accuracy, robustness, executability and real-time performance compared with two-stage pipelines. It can also scale to adapt to diverse humanoid embodiments, offering a new scheme for convenient, large-scale acquisition of multi-robot motion data.

Limitations. Constrained by model capacity and training data distribution, our method can still exhibit errors in global trajectory estimation for large-range human motions or when the camera is noticeably moving. In addition, our current method do not cover complex terrains or contact-rich human–environment interaction scenarios. Future work could incorporate more diverse data and model scene geometry and physical contacts, thus enabling robots to directly learn more challenging, scene-conditioned motion skills from videos.

REPRODUCIBILITY STATEMENT

The main text of the paper introduces the model architecture, supervision objectives, loss functions, evaluation metrics, and experimental comparison results. Appendix further supplements the implementation details. The source code and configuration files will be released on GitHub, with accompanying documentation describing how to obtain the official dataset and related resources. These materials are intended to enable researchers to independently reproduce the experimental procedures described in this paper.

AI USE STATEMENT

Generative AI tools were used only for limited language assistance, including wording suggestions, edits for readability, and translation. Any information suggested by these tools was checked by the authors against the original sources before inclusion, and all AI-assisted text was reviewed, revised as needed, and approved by the authors. The research questions, technical contributions, methodology, experiments, analyses, and conclusions were developed and carried out by the authors. The authors take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics*, 39 (4):62, 2020. doi: 10.1145/3386569.3392462.

Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. In *Proceedings of The Conference on Robot Learning*, Proceedings of Machine Learning Research, 2025.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics*, 24 (3):408–416, 2005. doi: 10.1145/1073204.1073207.

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Ko Ayusawa and Eiichi Yoshida. Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization. *IEEE Transactions on Robotics*, 33 (6):1343–1357, 2017. doi: 10.1109/TRO.2017.2752711.

Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8726–8737, 2023.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, pp. 561–578. Springer, 2016. doi: 10.1007/978-3-319-46454-1_34.

Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3D human pose and shape from a video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1964–1973, 2021.

Kourosh Darvish, Yeshasvi Tirupachuri, Giulio Romualdi, Lorenzo Rapetti, Diego Ferigo, Francisco Javier Andrade Chavez, and Daniele Pucci. Whole-body geometric retargeting for humanoid robots. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 679–686, 2019. doi: 10.1109/humanoids43949.2019.9035059.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14783–14794, 2023. doi: 10.1109/ICCV51070.2023.01358.

Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and inferring dense full-body human-scene contact. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13274–13285, 2022.

Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics*, 37(6):185:1–185:15, 2018. doi: 10.1145/3272127.3275108.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. doi: 10.1109/tpami.2013.248.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7122–7131, 2018. doi: 10.1109/CVPR.2018.00744.

Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5614–5623, 2019. doi: 10.1109/cvpr.2019.00576.

Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5253–5263, 2020. doi: 10.1109/CVPR42600.2020.00530.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. BFM-Zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. *arXiv preprint arXiv:2511.04131*, 2025. URL https://arxiv.org/abs/2511.04131.

Minas Liarokapis, Kostas J Kyriakopoulos, and Charalampos Bechlioulis. Directions, methods and metrics for mapping human to robot motion with functional anthropomorphism: A review. *Technical Report, Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, Greece.*, 2013.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, 2015. doi: 10.1145/2816795.2818013.

Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 33(6):220:1–220:13, 2014. doi: 10.1145/2661229.2661273.

Yi Lu, Shenghao Ren, Tianyu Xiong, Zhaoxiang Li, Jiaqi Li, He Zhang, Tao Yu, Qiu Shen, and Xun Cao. Pressmimic: Pressure-guided motion capture and control for humanoid robot imitation, 2026. URL https://arxiv.org/abs/2606.26741.

Zhengyi Luo, S. Alireza Golestaneh, and Kris M. Kitani. 3d human motion estimation via motion compression and refinement. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. SONIC:

Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11 (117):eaed4592, 2026. doi: 10.1126/scirobotics.aed4592.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. URL https://amass.is.tue.mpg.de.

Shin'ichiro Nakaoka, Atsushi Nakazawa, Kazuhito Yokoi, Hirohisa Hirukawa, and Katsushi Ikeuchi. Generating whole body motions for a biped humanoid robot from captured human dances. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pp. 3905–3910, 2003. doi: 10.1109/ROBOT.2003.1242196.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10975–10985, 2019. doi: 10.1109/cvpr.2019.01123.

Nancy S. Pollard, Jessica K. Hodgins, Marcia J. Riley, and Christopher G. Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pp. 1390–1397, 2002. doi: 10.1109/robot.2002.1014737.

Shenghao Ren, Yi Lu, Jiayi Huang, Jiayi Zhao, He Zhang, Tao Yu, Qiu Shen, and Xun Cao. MotionPRO: Exploring the role of pressure in human MoCap and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27760–27770, 2025.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. doi: 10.1145/3130800.3130883.

Constant Roux, Ludovic De Matteïs, Armand Jordana, Valentin Guillet, Nicolas Mansard, Olivier Stasse, and Philippe Souères. Direct dynamic retargeting for humanoid imitation learning from videos, 2026. URL https://arxiv.org/abs/2605.23762.

H. Martin Schepers, Daniel Roetenberg, and Peter H. Veltink. Ambulatory human motion tracking by fusion of inertial and magnetic sensing with adaptive actuation. *Medical & Biological Engineering & Computing*, 48(1):27–37, 2010. doi: 10.1007/s11517-009-0562-9.

Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11. ACM, 2024. doi: 10.1145/3680528.3687565.

Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J. Black. WHAM: Reconstructing world-grounded humans with accurate 3D motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2070–2080, 2024.

Aleš Ude, Christopher G. Atkeson, and Marcia Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47(2–3):93–108, 2004. doi: 10.1016/j.robot.2004.03.004.

Vicon Motion Systems Ltd. Vicon motion capture systems, 2026. URL https://www.vicon.com/. Accessed: 2026-09-20.

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8639–8648, 2018. doi: 10.1109/CVPR.2018.00901.

Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3D human shape and pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13033–13042, 2021.

Qingrui Zhao, Kaiyue Yang, Xiyu Wang, Shiqi Zhao, Yi Lu, Xinfang Zhang, Qiu Shen, Xiao-Xiao Long, and Xun Cao. Make tracking easy: Neural motion retargeting for humanoid whole-body control, 2026. URL https://arxiv.org/abs/2603.22201.

APPENDIX

A SEMANTIC KEYPOINT SPECIFICATION

The unified motion supervision uses an ordered semantic skeleton with 18 keypoints. The pelvis is the root. And every nonroot keypoint and its parent define one directed link for morphology alignment. Table 5 gives the SMPL joint associated with each keypoint and its position in the semantic hierarchy.

Table 5: Definition of the 18 semantic keypoints shared by the human and robot motion representations.

Idx.	Parent	SMPL	keypoints	Idx.	Parent	SMPL	keypoints
0	–	0	pelvis	9	0	6	torso
1	0	1	left hip	10	9	16	left shoulder
2	1	4	left knee	11	10	18	left elbow
3	2	7	left ankle	12	11	20	left wrist
4	3	10	left foot	13	12	22	left hand
5	0	2	right hip	14	9	17	right shoulder
6	5	5	right knee	15	14	19	right elbow
7	6	8	right ankle	16	15	21	right wrist
8	7	11	right foot	17	16	23	right hand

The same keypoint order is used throughout training and evaluation. On the human side, the keypoints are selected from the annotated SMPL body and transformed by morphology alignment. On the robot side, we extract the corresponding joints as the mapping keypoints. Foot refinement leverages the configured foot keypoints for adjustment. When a foot mesh model is available, the positions of the bottom points will be used to represent the sole of the foot.

B TRAINING OBJECTIVE DEFINITIONS

This section defines the root, temporal, and joint-limit terms in Eq. (6), followed by the separately trained contact objective. Throughout the following loss and metric definitions, let $\Delta_t \mathbf{z}_t = \mathbf{z}_t - \mathbf{z}_{t-1}$ denote the backward finite difference and let ν be the frame rate in hertz. We use r and d_r for the target robot and its joint DoF count, as in Eq. (1), and omit the robot index on semantic keypoints as in Eq. (3). The orientation matrix of $\hat{\mathbf{o}}_t^{(r)} \in \mathbb{R}^6$ is $\hat{\mathbf{R}}_t^{(r)} \in \mathrm{SO}(3)$. A superscript $\star$ denotes the corresponding ground-truth quantity. For a scalar ξ, the implementation uses $\mathrm{sl1}(\xi) = \xi^2/2$ when $|\xi| < 1$ and $\mathrm{sl1}(\xi) = |\xi| - 1/2$ otherwise; ρ averages this penalty over vector coordinates as defined below Eq. (4). The root losses are

$$\begin{aligned} \mathcal{L}_{\mathrm{trans}} &= \frac{1}{n}\sum_{t=1}^{n} \rho(\hat{\mathbf{p}}_t - \mathbf{p}_t^{\star}), \\ \mathcal{L}_{\mathrm{ori}} &= \frac{1}{n}\sum_{t=1}^{n} \mathrm{dist}_{\mathrm{SO}(3)}(\hat{\mathbf{R}}_t^{(r)}, \mathbf{R}_t^{(r),\star}), \\ \mathrm{dist}_{\mathrm{SO}(3)}(\mathbf{R}_1, \mathbf{R}_2) &= \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_1^{\top}\mathbf{R}_2) - 1}{2}\right). \end{aligned} \tag{9}$$

$\mathrm{dist}_{\mathrm{SO}(3)}$ is the geodesic distance between two rotation matrices and is measured in radians; the implementation clamps its cosine argument for numerical stability. Translation is measured in meters.

Velocities and accelerations use $\nu\Delta_t \mathbf{z}_t$ and $\nu^2\Delta_t^2 \mathbf{z}_t$. For root localized keypoints, define

$$\hat{\mathbf{x}}_{t,j}^{\mathrm{loc}} = (\hat{\mathbf{R}}_t^{(r)})^{\top}(\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_{t,0}), \qquad \mathbf{y}_{t,j}^{\mathrm{loc}} = (\mathbf{R}_t^{(r),\star})^{\top}(\mathbf{y}_{t,j} - \mathbf{y}_{t,0}). \tag{10}$$

Let $\mathcal{J}_{\mathrm{dist}} \subseteq \mathcal{J}$ index the selected distal keypoints (both feet, wrists, and hands for G1). The temporal term used by the released configuration is

$$\mathcal{L}_{\mathrm{temp}} = \beta_{\mathrm{ev}}\mathcal{L}_{\mathrm{ev}} + \beta_{\mathrm{ea}}\mathcal{L}_{\mathrm{ea}} + \beta_{\mathrm{smooth}}\mathcal{L}_{\mathrm{smooth}} + \beta_{\mathrm{jit}}\mathcal{L}_{\mathrm{jit}}, \tag{11}$$

where $\mathcal{L}_{\text{ev}}$ and $\mathcal{L}_{\text{ea}}$ are mean Smooth L_1 velocity and acceleration discrepancies between $\hat{\mathbf{x}}^{\text{loc}}_{t,j}$ and $\mathbf{y}^{\text{loc}}_{t,j}$ for $j \in \mathcal{J}_{\text{dist}}$. $\mathcal{L}_{\text{smooth}}$ is a weighted sum of the second difference magnitudes of $\hat{\mathbf{p}}_t$, $\hat{\mathbf{o}}^{(r),\text{6D}}_t$, and $\hat{\mathbf{q}}^{(r)}_t$, each divided by a fixed positive reference scale before combination. $\mathcal{L}_{\text{jit}}$ is the mean third difference magnitude of all $\hat{\mathbf{x}}^{\text{loc}}_{t,j}$. The coefficients β_{ev}, β_{ea}, β_{smooth}, and β_{jit} are nonnegative scalar weights. This normalization prevents quantities in meters, radians, and the dimensionless 6D rotation representation from being mixed without scale control.

For joint limits, let $\mathbf{q}^{(r)}_{\min}, \mathbf{q}^{(r)}_{\max} \in \mathbb{R}^{d_r}$ be the lower and upper angle vectors in radians. The penalty is

$$\mathcal{L}_{\text{lim}} = \frac{1}{nd_r} \sum_{t=1}^{n} \left(\|[\mathbf{q}^{(r)}_{\min} - \hat{\mathbf{q}}^{(r)}_t]_+\|_2^2 + \|[\hat{\mathbf{q}}^{(r)}_t - \mathbf{q}^{(r)}_{\max}]_+\|_2^2 \right). \tag{12}$$

Here $[\cdot]_+$ denotes the elementwise positive part. Data-fitting and temporal losses use valid-frame masks; temporal differences require all participating frames to be valid. The joint-limit regularizer averages over all output frames and joints.

Contact prediction loss. With the motion estimator frozen, the contact predictor learns the pressure-derived labels $c^\star_{t,f}$ for the two feet. Its objective is

$$\mathcal{L}_{\text{contact}} = \frac{1}{|\mathcal{V}|} \sum_{(t,f)\in\mathcal{V}} \text{BCE}\big(\hat{c}_{t,f}, c^\star_{t,f}\big), \tag{13}$$

where $\mathcal{V}$ contains valid frame–foot pairs and BCE is binary cross-entropy. This loss trains the contact predictor separately from $\mathcal{L}_{\text{motion}}$; its architecture and training settings are given in Appendix F.

C Evaluation metric definitions

For root trajectory evaluation, one rigid transform is fitted to the complete sequence:

$$\gamma = \arg\min_{\gamma\in\text{SE}(3)} \sum_{t=1}^{n} \|\gamma(\hat{\mathbf{p}}_t) - \mathbf{p}^\star_t\|_2^2. \tag{14}$$

For pose evaluation, $\tilde{\mathbf{x}}_{t,j} = \phi_t(\hat{\mathbf{x}}_{t,j}, \mathbf{y}_{t,j})$ is the Procrustes-aligned robot keypoints defined in the main text. The transform fits a rotation, translation, and uniform scale independently per frame. It is distinct from the root-local keypoints $\hat{\mathbf{x}}^{\text{loc}}_{t,j}$ in Eq. (10). The principal pose and temporal metrics are

$$\begin{aligned}
\text{MAMPJPE} &= \frac{10^3}{n|\mathcal{J}|} \sum_{t=1}^{n} \sum_{j\in\mathcal{J}} \|\tilde{\mathbf{x}}_{t,j} - \mathbf{y}_{t,j}\|_2, \\
\text{RTE} &= \frac{10^3}{n} \sum_{t=1}^{n} \|\gamma(\hat{\mathbf{p}}_t) - \mathbf{p}^\star_t\|_2, \\
\text{Jitter} &= \frac{\nu^3}{10(n-3)|\mathcal{J}|} \sum_{t=4}^{n} \sum_{j\in\mathcal{J}} \big\|\Delta^3_t \hat{\mathbf{x}}_{t,j}\big\|_2, \\
\text{Accel} &= \frac{10^3}{(n-2)|\mathcal{J}|} \sum_{t=3}^{n} \sum_{j\in\mathcal{J}} \big\|\Delta^2_t \hat{\mathbf{x}}_{t,j} - \Delta^2_t \mathbf{y}_{t,j}\big\|_2.
\end{aligned} \tag{15}$$

Slide uses the same external stable support mask for every method. The mask is derived from pressure contact labels and the reference G1 foot motion, with a horizontal speed threshold of 1 m/s and removal of support segments shorter than four frames. Slide is the mean horizontal displacement between adjacent stable support frames in millimeters per frame. Violation counts contact foot–frame samples more than 20 mm above or 30 mm below a reference height. For each foot and sequence, this reference is the median predicted foot-keypoints height over the evaluation contact mask; single-frame contact runs are excluded from the violation count. TDE is the pelvis to torso direction error, and EDE averages the two wrist to hand and two ankle to foot direction errors; both are reported in degrees.

Simulation protocol. For simulation evaluation, we use the standard SONIC controller (Luo et al., 2026) in MuJoCo with the same G1 model and controller settings for all seven methods in Table 1. The 12 source sequences at 30 FPS are divided into nonoverlapping, approximately equal clips of about 600 frames, covering every source frame. Identical boundaries are used for all methods and human ground truth, yielding 362 clips per method. References are resampled to 50 Hz, and each rollout is independently reset to its reference initial pose.

Executed success rate. At motion start, the tracking reference is aligned once to the actual robot's initial heading and horizontal position. Following a 0.5 s grace period, tracking is checked at 50 Hz. A rollout terminates when a threshold violation persists for three consecutive samples: root height error above 0.25 m, full 3D root orientation error above 1.0 rad, or maximum height error across the two ankles and two wrists above 0.25 m. Safety stops, premature deployment exits, initialization failures, and timeouts exceeding the expected motion duration by 3 s also count as failures. The default fall detector is disabled; success requires an explicit motion-completion signal. Thus, with n_{succ} completed clips and $n_{\mathrm{exec}} = 362$ evaluated clips per method,

$$\text{Exec. SR} = \frac{n_{\mathrm{succ}}}{n_{\mathrm{exec}}}. \tag{16}$$

Executed-motion accuracy. Exec. MAMPJPE compares the robot's actual simulated motion with human ground truth. We reconstruct SMPL joints from the annotated shape, pose, and translation, apply the same 18-keypoints morphology mapping as above, and resample the targets to the rollout timestamps. Let $\mathcal{T}$ denote the valid recorded frames of a clip and $\tilde{\mathbf{x}}_{t,j}^{\mathrm{sim}}$ its executed robot keypoints after per-frame similarity Procrustes alignment to $\mathbf{y}_{t,j}$. This alignment fits rotation, translation, and uniform scale, as in Eq. (15). The clip error in millimeters is

$$\text{Exec. MAMPJPE} = \frac{10^3}{|\mathcal{T}|\,|\mathcal{J}|} \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \left\| \tilde{\mathbf{x}}_{t,j}^{\mathrm{exec}} - \mathbf{y}_{t,j} \right\|_2 . \tag{17}$$

We aggregate clip errors by their valid-frame counts, including recorded prefixes of failed rollouts. A clip with no valid metric frames contributes to the success-rate denominator but not to the error average.

D Experimental protocol and aggregation

Data and supervision. MotionPRO (Ren et al., 2025) provides synchronized RGB images, optical motion capture, and pressure measurements. RGB is the only input to the motion predictor. Optical motion capture provides the human pose and root motion used to construct supervision in the unified motion space. Pressure measurements are used only to generate binary contact labels for the left and right feet. Neither pressure nor human motion is required at inference time, and no paired robot joint trajectory is used for training.

Sequences are divided by performer, so the same person does not appear in both the training and test sets. G1 is used to train the shared motion representation and the complete robot motion decoder. For each additional robot, the visual temporal encoder, MLP trunk, and pretrained root translation branch are reused without further optimization. Its robot-specific joint branch and root rotation branch are trained with the same supervision losses and adapter-specific weights after configuring the robot kinematic tree, semantic keypoints mapping, and morphology scales.

Baseline protocol. The estimated input baselines first recover human motion with GVHMR or WHAM and then apply NMR or GMR. The Oracle baselines replace the estimated human motion with its ground truth value, which isolates the error introduced by human to robot mapping. All G1 comparisons use the same evaluation sequences, frame rate, robot kinematics, semantic keypoints definitions, and output coordinate convention (Humans and robots are unified under the same rotation frame.). Every robot trajectory is converted to the shared semantic keypoints geometry through the same forward kinematics procedure before evaluation.

Table 6: Cross dataset robot motion capture results on RICH. All methods are evaluated on the same 199 continuous clips from 194 camera sequences, with 120,709 frames and 524 evaluation windows, using annotation-assisted target localization. Lower is better for all metrics. MAMPJPE is in millimeters, and Jitter follows Appendix C. Red and yellow mark the best and second best results, respectively.

Method	Clips	Frames	MAMPJPE ↓	Fail_{100} ↓	Jitter ↓
BeyondRetarget	199	120,709	67.970	8.78%	4.266
GVHMR→NMR	199	120,709	65.745	9.35%	4.284
GVHMR→GMR	199	120,709	70.020	20.04%	16.411

E Cross dataset evaluation on RICH

We conduct a cross-dataset evaluation on G1 using the RICH validation split (Huang et al., 2022), comparing BeyondRetarget with GVHMR followed by NMR or GMR. All methods use a common subset with available camera calibration and body annotations and projected target box centers inside the image. Splitting at temporal gaps yields 199 continuous clips from 194 camera sequences, totaling 120,709 frames.

Subject, camera, and frame identifiers associate each image with its target annotation. Following the RICH toolkit's projection, we obtain person-specific boxes from the annotated gendered SMPL-X meshes and use them for all methods. This provides annotation-assisted target localization; ground truth body parameters are not otherwise inputs to the motion predictors. All methods are rerun on these crops.

For evaluation, gendered SMPL-X meshes are mapped to SMPL and regressed to joints, then converted to our common gravity-aligned G1 semantic-keypoints reference. Table 6 reports MAMPJPE, Fail_{100}, and Jitter following Appendix C.

BeyondRetarget remains close to the best two-stage method GVHMR→NMR in cross-dataset pose accuracy under distribution shift: its MAMPJPE is only 2.225 mm higher. Meanwhile, among the three reported methods, it achieves the best Fail_{100} and Jitter, with values of 8.78% and 4.266, demonstrating that this direct prediction method retains strong robustness and temporal smoothness under distribution shift.

When viewed in the context of the current training data and model design, our robot motion predictor is trained on the indoor camera-stationary MotionPRO dataset with limited scene and motion diversity, and adopts a relatively compact model architecture. In contrast, GVHMR is trained on a substantially broader mixture of AMASS, BEDLAM, Human3.6M, and 3DPW. Insufficient coverage of the training distribution and limited model capacity account for the modest performance gap relative to the best two-stage method on completely unseen RICH scenes. Expanding the distribution of robot motion training data and upgrading the model architecture serve as a viable path to further improve the generalization of our method. We expect these improvements to effectively enhance the model's coverage of unseen scene conditions and yield substantial performance gains over the two-stage method across all scenarios.

Experimental results show that even under the current training-data and model-capacity setting, this method can achieve average pose accuracy close to that of the best two-stage methods while maintaining the lowest high-error failure rate, which highlights the value of the BeyondRetarget framework. It also demonstrates that the end-to-end prediction framework is a technical route with greater potential for improvement. Expanding the dataset and scaling up the model can further optimize the generalization ability and motion estimation accuracy of the BeyondRetarget framework.

F Architecture and training details

Visual and temporal encoding. The visual front end uses features extracted by a frozen HMR2 encoder (Goel et al., 2023). The shared temporal encoder (Lu et al., 2026) contains a two layer bidirectional GRU, two temporal attention layers with rotary position encoding, and attention across windows. Offline inference uses windows of 50 frames with an overlap of 10 frames. Each window can attend to compressed information from as many as eight neighboring windows in either temporal direction.

Robot motion decoder and optimization. The shared motion feature $\mathbf{u}_t$ is passed to the robot motion decoder Ψ_r, whose root translation and root rotation branches predict root motion and whose robot-specific joint branch predicts robot joint motion. During robot adaptation, the robot-specific joint branch and root rotation branch are optimized, while the visual temporal encoder, MLP trunk, and pretrained root translation branch are reused without further optimization. Table 7 lists the number of DoF predicted by each robot specific joint branch.

Contact predictor and optimization. The contact predictor combines the detached 1024-dimensional motion and visual features. Separate MLP adapters use a 256-dimensional hidden layer and produce 1024-dimensional motion and 256-dimensional visual embeddings. After layer normalization and concatenation, two 256-unit hidden layers produce two foot contact logits. Hidden layers use GELU and dropout of 0.1. All contact modules are optimized jointly with Eq. (13), while the motion estimator remains frozen. The released defaults use AdamW for 40 epochs, 64 windows per batch, learning rate 10^{-4}, and weight decay 10^{-4}. The best checkpoint is selected by the reported validation contact-F1 (averaged across feet and evaluation batches) at a 0.5 probability threshold. The refinement thresholds below serve a different purpose: temporal stabilization of the predicted contact states.

Table 7: Number of joint degrees of freedom predicted by each robot-specific joint branch.

Robot	DoFs	Robot	DoFs
G1	29	R1	26
GR1-T1	32	GR2-V3	29
H1	21	T1	23
Tienkung	20	Atlas	30

G CONTACT-AWARE REFINEMENT

Contact states and support selection. The following parameters describe the released offline refinement defaults at 30 FPS. For each foot, a hysteresis state $h_{t,f}$ is initialized to zero and updated from the predicted probability as

$$h_{t,f} = \begin{cases} 1, & \hat{c}_{t,f} \geq 0.60, \\ 0, & \hat{c}_{t,f} < 0.35, \\ h_{t-1,f}, & \text{otherwise.} \end{cases} \tag{18}$$

We first fill noncontact gaps of at most five frames that are bounded by contact on both sides, then remove contact runs shorter than four frames. The resulting $c_{t,f}$ is the binary contact state in Eq. (7). These operations regularize predictions at inference time and do not modify the pressure labels used to train the contact head.

Support additionally requires horizontal foot speed at most $\tau_v = 1.0$ m/s. Offline speed gating uses the larger of the incoming and outgoing adjacent-frame speeds, computed from forward kinematics; support runs shorter than four frames are then removed.

Offline root correction and inverse kinematics. We first smooth root translation, root rotation, and joint trajectories using third-order Savitzky–Golay filters , and renormalize the filtered quaternions. We then follow the two-level state decision in Eq. (7), contact activates root height correction, while stable support additionally enables root horizontal correction and lower-body inverse kinematics (IK). These three steps operate on the filtered motion predictions.

Root height correction. Forward kinematics provides the contacting foot positions, together with the robot's sole geometry when available. We use these positions to estimate the vertical root adjustment needed to bring the contacting soles toward the ground plane, reducing penetration and floating. Contact-based height references further stabilize the root: the stabilization offset is clipped to ± 4 cm and applied with a gain of 0.75. This step does not constrain horizontal motion, so a contacting foot can still slide when it is not in the support state.

Root horizontal correction. For stable supporting feet, we compensate their horizontal displacement by adjusting the root translation. The displacement correction uses nine-frame smoothing and a component-wise bound of 2 cm. To limit accumulated drift, horizontal anchors are maintained within contiguous support segments, with the resulting root stabilization offset bounded by 15 cm. Six-frame boundary ramps gradually introduce and remove the stabilization correction at support transitions.

Lower-body inverse kinematics. After root correction, each supporting foot is anchored to its position at the start of the support segment, and the target trajectory is smoothed over four frames. Starting from the filtered joint prediction, constrained cyclic coordinate descent (CCD) traverses each leg chain from the distal joints toward the hip. For each revolute joint, the solver computes the signed angle between the current and target joint-to-foot vectors about the joint's world-space axis. The update uses a step factor of 0.10, attenuated by the joint's chain position and support weight. Forward kinematics is recomputed after each update, joint angles are clamped to their limits, and the root pose remains fixed. The solver performs at most two sweeps, stopping early when the foot-position error falls below 1 mm. An ankle correction additionally aligns the supporting foot normal with the ground normal, using a gain of 0.6 and a maximum angular step of 0.16 rad. Finally, as in Eq. (8), the IK residual is smoothed with a five-frame moving average and forward and backward rate limiting of 0.18 rad per frame, then added to the prediction with joint-limit clamping.

H Streaming Implementation

Windowed prediction and streaming scheduling. The model is a sequence-to-sequence model: a window of 50 visual features produces 50 robot states and 50 pairs of contact probabilities. The offline runner advances by 40 frames (10-frame overlap), combines overlapping predictions with triangular weights, and uses up to eight context windows on either side.

Live inference reuses this network in a rolling-window pipeline, without waiting for observations beyond the current window. In the default live configuration, after an initial 50-frame buffer, every four new frames trigger one forward pass with batch size one window. The network still computes all 50 outputs, but only the last four previously uncommitted frames are emitted. For a window ending at t, these outputs correspond to $t-3, \ldots, t$; the next window ends at $t+4$. Cross-window context contains mean visual features from at most eight previously processed windows, with missing part masked out. The bidirectional GRU and attention within the observed window are retained.

Asynchronous live pipeline. For Table 4, all methods use an asynchronous pipeline with separate workers for decoding, detection, visual encoding, motion prediction, and action output. Models remain resident, and GPU stages use separate CUDA streams. All methods share a YOLO detector and box smoothing. The two-stage baselines use FP16 visual inference with ViTPose, and keep GVHMR inference and its official postprocessing in FP32. GVHMR uses a static-camera setting with image-size-based intrinsics. Human prediction overlaps with ordered retargeting in a persistent worker: GMR processes only newly committed frames, while NMR retains its temporal processing and filtering. All methods process one observed 50-frame window at a time and commit only the newest stride of outputs; no offline feature cache is used.

We select the latency-oriented and throughput-oriented settings, respectively, through bounded batch/step size search. We denote each setting by $(B_{\mathrm{det}}, B_{\mathrm{vis}}, S)$: the numbers of frames batched for YOLO detection and visual encoding (Goel et al., 2023), and the stride of motion predictions, respectively. The motion model processes one 50-frame window per call; GVHMR additionally requires predicted ViTPose keypoints, whose frame batch is selected together with the HMR2 batch. In the experimental settings, both use B_{vis} frames. For latency in Table 4, we use $(2, 2, 2)$ for BeyondRetarget and $(2, 4, 16)$ for both baselines, with individual FIFO action postprocessing and emission. For maximum throughput, we use $(2, 2, 4)$ and $(2, 16, 32)$, respectively, with action-processing batches of up to 64 for all methods. These are separate operating points. Both tests disable frame dropping and output skipping; unpaced input uses bounded queues with backpressure.

Online controller output. The optional ZMQ publisher reorders joints to the GR00T/IsaacLab convention, converts root rotations to normalized quaternions, and resamples cached actions. By default it publishes a 50-pose payload at 50 Hz, computing joint velocities from the resampled poses

at that rate. Samples beyond the latest available action hold the latest pose. The payload therefore does not represent 50 independently predicted future actions, and its publication rate is distinct from neural inference throughput.

Runtime measurement. All methods are measured on the same NVIDIA RTX 3090, with two runs per setting after model warm-up. For latency, frames arrive at a scheduled 30 Hz, and timestamps span from scheduled arrival before decoding to completed robot-action output after method-specific postprocessing. The reported median pools the same steady-state frame interval across both runs, excluding initial context accumulation and the final tail. Throughput instead uses unpaced input and output, and reports the mean number of emitted robot actions per elapsed pipeline second. The reported VRAM is the sampled peak simultaneous GPU memory of the inference and retargeting processes in the latency setting. These results characterize the evaluated rolling-window implementations rather than an architectural performance limit.

I METHOD CLARIFICATION

The core innovation of this work is not simply to combine human motion capture and retargeting modules, but to change the mechanism for robot motion generation. Conventional approaches first obtain an explicit SMPL motion and then map it to the target robot, whereas BeyondRetarget directly predicts robot motion from visual input. Therefore, the terms *direct* and *end-to-end* in this work emphasize a change in the intermediate motion representation used during inference: robot motion is no longer generated through an explicit human motion intermediate. This change allows robot motion prediction to directly exploit motion information from the original visual input and shifts the overall objective from "recovering human motion" to "generating usable motion for the target robot."

The *shared motion representation* serves as the key interface in this formulation. It is a latent motion representation learned from video, rather than a unified physical joint space for humans and robots or a conventional human-to-robot retargeting space. Its role is to preserve motion information that can be shared across different morphologies and map it to the specific motions of different humanoids through robot-specific decoders. Human body models and semantic keypoint supervision are used during training to establish this correspondence, allowing human motion to supervise robot motion learning, while human model itself is not required as an explicit intermediate representation during inference. Therefore, the use of human motion supervision does not reduce the method to a conventional two-stage pipeline; instead, it enables visual motion information to be directly learned into the cross-morphology structure required for robot motion generation.

The term *executable* in this work also has a concrete experimental meaning. It does not claim that the generated motions are fully dynamically feasible for arbitrary robots, environments, or control conditions. Rather, it refers to robot reference motions that have been experimentally executed and validated on the simulation controller and physical robot platforms used in this work. The 95.86% executed success rate, successful execution of multiple motions on a real robot, and the substantial improvements in trajectory, jitter, acceleration, foot sliding, and violation brought by contact-aware refinement together show that *executable* is not merely another description of static pose accuracy, but reflects practical execution capability involving motion continuity and support stability.

The robustness of our method should likewise not be understood merely as lower average error. BeyondRetarget achieves 0% Fail_{65} and 0% Fail_{100} over the evaluated windows, whereas all evaluated two-stage methods contain high-error failure windows. This indicates that, across the diverse motions covered by the evaluation, including challenging cases with large-amplitude motions and self-occlusion, the method does not exhibit the severe motion collapse or numerical breakdown commonly observed in two-stage pipelines. Furthermore, even when ground-truth human motion is provided to the two-stage methods, their MAMPJPE results remain inferior to BeyondRetarget, indicating that the error sources in the conventional pipeline arise not only from human motion estimation but also from the subsequent human-to-robot mapping process. BeyondRetarget therefore does not merely achieve lower average motion error, but also substantially reduces the occurrence of severe failures while preserving motion quality.

The multi-robot adaptation capability arises from the joint design of the shared motion representation and unified supervision scheme. The former extracts visual motion information that can be

shared across different robot morphologies, while the latter establishes correspondence between human motion and different robots through consistent semantic keypoints and morphology-aware alignment. Based on this design, only the robot-specific branches need to be trained according to the kinematic structure of the target robot, allowing the same motion representation to be mapped to different humanoid embodiments. We validate this adaptation process on eight humanoid embodiments, with approximately one hour required to adapt a new robot, demonstrating efficient expansion across different robot morphologies. Here, the multi-robot capability refers to efficient adaptation based on the shared representation rather than zero-shot prediction for arbitrary unseen robots.

Meanwhile, enabled by the direct prediction of the end-to-end approach and the simplicity of the model design, our streaming inference reaches 50.02 FPS with a median camera-to-command latency of 192.8 ms. Its real-time performance substantially exceeds that of the two-stage methods, further demonstrating the real-time potential of the proposed end-to-end architecture.

Taken together, the experiments validate different properties of the proposed framework from multiple complementary perspectives: morphology-aligned pose and trajectory metrics evaluate motion reproduction; Jitter, Accel, Foot Sliding, and Violation evaluate temporal continuity and support consistency; high-error failure statistics and cross-dataset experiments evaluate robustness; simulation and real-robot execution evaluate executability; eight humanoid robots evaluate cross-morphology adaptation; and latency, throughput, and memory measurements evaluate practical runtime efficiency. These experiments therefore do not merely demonstrate that a particular network architecture can achieve favorable motion metrics, but jointly validate the effectiveness of this new formulation, which differs from the conventional "human motion reconstruction $\rightarrow$ retargeting" pipeline, across motion quality, robustness, robot execution, multi-morphology adaptation, and real-time performance.

The core value of this formulation lies in the fact that it no longer relies on explicit human motion reconstruction and subsequent retargeting during inference, while still being able to exploit rich human motion annotations for training and directly output robot motion trajectories. This reduces the information access limitations introduced by a fixed human intermediate representation and the error accumulation caused by multi-stage mapping, while making the resulting robot motions directly available for evaluation, execution, and reuse and enabling efficient adaptation to different robot morphologies. Combined with the accessibility and large scale of monocular video, BeyondRetarget therefore provides a new path for robot motion acquisition: large volumes of human monocular videos can be directly transformed into high-quality, executable humanoid robot motion data, providing a scalable demonstration acquisition scheme for robot imitation learning, motion learning, and subsequent whole-body control. At the same time, the real-time performance enabled by the end-to-end inference framework and its simple model design also makes it applicable to latency-sensitive downstream tasks such as real-time visual teleoperation.

J REPRODUCIBILITY DETAILS

Data preparation. We use the MotionPRO training and evaluation splits, separated by performer so that a performer does not occur in both splits. Each sequence is processed from the front-view video at its native frame order and a nominal rate of 30 FPS. The preprocessing pipeline generates person bounding boxes with the YOLOv8x tracker, extracts 1024-dimensional HMR2 visual tokens, converts the provided SMPL parameters to the first 24 SMPL joints with the neutral SMPL model, and derives left/right forefoot contact labels from the pressure maps and SMPL foot positions. The generated boxes, visual tokens, joints, contacts, and SMPL records are required to have the same number and order of frames.

Inference and evaluation. For each sequence, the model predicts root translation, root orientation, robot joint values, and foot-contact probabilities in the same frame order as the input features. Offline post-processing first applies Savitzky–Golay filtering to the root and joint trajectories, then converts contact probabilities to binary states with hysteresis, fills short gaps, removes short contact segments, and identifies stable support using foot horizontal speed. Contact frames enable sole-height and vertical-root correction; stable support additionally enables bounded horizontal correction and lower-body inverse kinematics. The final motion is exported through the robot-specific kinematic model, and all metrics use the aggregation rules in Appendix C and Appendix D. Stream-

ing inference reuses the windowed model with past-only cross-window context and, by default, commits only the newest four frames. Its within-window lookahead, online stabilization, and latency definition are described in Appendix G.